Entity-centric Data Fusion on the Web

Andreas Thalhammer
Institute AIFB, Karlsruhe Institute of Technology
andreas.thalhammer@kit.edu

Steffen Thoma
Institute AIFB, Karlsruhe Institute of Technology
steffen.thoma@kit.edu

Andreas Harth
Institute AIFB, Karlsruhe Institute of Technology
andreas.harth@kit.edu

Rudi Studer
Institute AIFB, Karlsruhe Institute of Technology
rudi.studer@kit.edu

ABSTRACT

A lot of current web pages include structured data which can directly be processed and used. Search engines, in particular, gather that structured data and provide question answering capabilities over the integrated data with an entity-centric presentation of the results. Due to the decentralized nature of the web, multiple structured data sources can provide similar information about an entity. But data from different sources may involve different vocabularies and modeling granularities, which makes integration difficult. We present an approach that identifies similar entity-specific data across sources, independent of the vocabulary and data modeling choices. We apply our method along the scenario of a trustable knowledge panel, conduct experiments in which we identify and process entity data from web sources, and compare the output to a competing system. The results underline the advantages of the presented entity-centric data fusion approach.

CCS CONCEPTS

• **Information systems → Data extraction and integration**; **Resource Description Framework (RDF)**;

KEYWORDS

entity-centric data fusion; data/knowledge fusion; structured data; linked data; n-ary relations; entity data fusion; data provenance

1 INTRODUCTION

Between December 2014 and December 2015 the percentage of web pages that include semantic markup has risen from 22% to 31.3% [16]. A large fraction of the structured data is based on schema.org annotations, which can be parsed to RDF [23], a graph-structured data model specified by the W3C. Government initiatives, non-profit organizations, and commercial data providers publish structured data on the web. They often use data publication features of current content management or electronic shop systems. Some organizations even provide a dedicated interface on top of their databases following the Linked Data principles [2]. Large-scale retrieval systems (e.g., search engines) collect, clean, normalize, and integrate the data to drive user-facing functionality [10, 28].

Data from the web is heterogeneous, as pointed out in a recent paper [3], where "heterogeneity, quality and provenance" has been identified as one of the four most pressing topics concerning the Semantic Web: "It is a truism that data on the Web is extremely heterogeneous. [...] A dataset precise enough for one purpose may not be sufficiently precise for another." [3] As the data can differ in modeling granularities and can be sparse and overlapping across sources, integration on the quality level that supports fine-grained querying often requires manual curation to map the data to a canonical representation [34]. Less manual effort is involved in supporting entity-centric views of only parts of the data, as done in so-called "knowledge panels", which tolerate noisier data. The knowledge panels only contain a condensed top-k rendering of the data and use ranking to achieve high precision at k data items [28, 30]; k is often very small (10 or 20) compared to the overall available data for an entity (which can often be multiple GBs).[1] However, many knowledge panels often do not show the provenance for individual data items and doubts about correctness or notability have been pointed out [6, 14].

In this paper, we address the problem of *entity-centric data fusion*. In essence, we tackle the challenge of identifying when multiple sources make the same claim[2] about an entity in different structured ways (i.e., by using different RDF vocabularies)—which boils down to: different URIs, varying literals, and different *modeling granularities*. We leverage what could be perceived as "cross-source redundancy" by reconciling identical or similar claims while still keeping track of the respective sources and their representation of a claim.

Our approach is based on a data processing pipeline, which takes as input a set of equivalent entity identifiers and provides as output a similarity-based grouping (clustering) of RDF triples and *chains of triples* from multiple sources that describe the entity. The pipeline consists of the following steps: retrieve claims from different web sources; extract path features; perform hierarchical clustering; refine clusters; and select representatives. We provide placeholder steps around record linkage (at the start of the pipeline) and filtering/ranking (at the end), which can be implemented depending on the specific scenario. The focus of our approach is to establish mappings in entity-centric data while accounting for different modeling granularities. This also includes the mappings between the involved vocabulary terms and entity identifiers. The basic idea is to move back and forth between representing claims about an entity in a structured way (based on identifiers and triples) and representing claims as strings. In contrast to more traditional data

[1] The process of selecting the k most important data items about an entity is also called "entity summarization" [30, 31].

[2] We use the term "claim" when one or multiple sources state a concise piece of information in RDF, independent of its concrete modeling (in RDF), and the term "triple" when information is represented in a single subject-predicate-object notation.

HT'17, July 4-7, 2017, Prague, Czech Republic.
2017. 978-1-4503-4708-2/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3078714.3078717

integration methods (e.g., [11]), we do not directly aim at identifying contradicting information but our approach can be extended with such functionality. A straight-forward interpretation of the output of our method could be a weighting/ranking of claims about an entity in accordance to the number of sources that make it.

Generic or customized record linkage algorithms [18, 19] commonly solve the problem of establishing equality between entity identifiers. In our work, we assume that entity identifiers are already linked (as done via `owl:sameAs` in web data). In the past, many ontology alignment approaches relied on clean and extensively modeled ontologies without making strong use of instance data (e.g., [27]). For example, [26] mapped different modeling granularities between two extensively modeled ontologies using "complex correspondences" expressed in rules. On the web, with many different ontologies which are often inadequately modeled for ontology alignment purposes, we require a more robust method. Approaches such as [25, 29] allow for more heterogeneous input data, but do not address modeling granularities ([29] identifies "structural heterogeneity" to be addressed in future work). In fact, nowadays many sources, most prominently Wikidata [33], use n-ary relations for modeling RDF data in combination with additional context factors [13, 17], making the problem of addressing their integration more acute.

The contributions of our work are as follows:

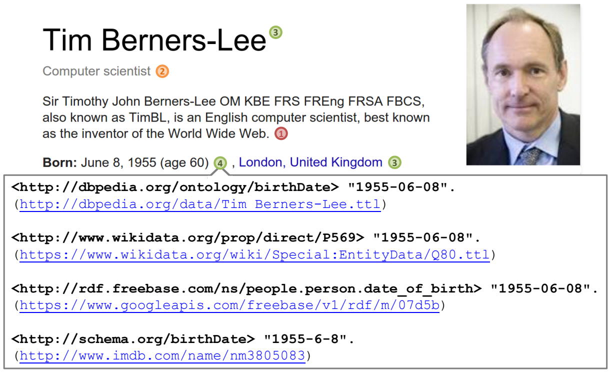
- We present the problem of granularity-agnostic entity data fusion for graph-structured data on the web.
- We provide an entity-centric approach that enables the fusion of claims from multiple web sources without prior knowledge about the used schemas or required alignment patterns, taking into account data provenance.
- We introduce the concept of path features, a graph reconciliation model that enables easy switching between (multi-hop) paths and their string representations.
- In our experiments, we measure the effectiveness of the entity data fusion approach for entity-centric, multi-sourced claims and demonstrate superiority over a baseline established as part of the Sig.ma system in [32].

2 EXAMPLE

Figure 1 depicts our idea of a "trustable knowledge panel". The colors of the buttons implement a traffic light scheme for the trustability of the claims. By clicking on such a button, a pop up would open that provides direct reference to documents which cover the claim together with each document's representation. Amongst others, such a panel can serve two important purposes:

(1) users can verify the sources provided for a claim; and
(2) the number of sources can serve as a straight-forward justification for notability.

The trustable knowledge panel requires the integration of data from multiple sources, both on the syntactic and semantic level. The input to the panel would be a single unique identifier for an entity (e.g., http://dbpedia.org/resource/Tim_Berners-Lee). Before rendering the panel, we require to have multiple groups of claims about the same entity. The claims can be single RDF triples or multiple ones (depending on the used vocabulary, in particular modeling granularities). The claims of one group can come from different sources. Each group has a representative. The groups can be ranked via "number of sources" but other ranking methods and combinations are possible. In our work, we focus on the steps in the pipeline to reconcile and fuse the data from multiple sources. The trustable knowledge panel serves as illustration of how to apply the output of our system in a user-facing scenario.



Tim Berners-Lee ③

Computer scientist ②

Sir Timothy John Berners-Lee OM KBE FRS FREng FRSA FBCS, also known as TimBL, is an English computer scientist, best known as the inventor of the World Wide Web. ①

Born: June 8, 1955 (age 60) ④ , London, United Kingdom ③

```
<http://dbpedia.org/ontology/birthDate> "1955-06-08".
(http://dbpedia.org/data/Tim_Berners-Lee.ttl)

<http://www.wikidata.org/prop/direct/P569> "1955-06-08".
(https://www.wikidata.org/wiki/Special:EntityData/Q80.ttl)

<http://rdf.freebase.com/ns/people.person.date_of_birth> "1955-06-08".
(https://www.googleapis.com/freebase/v1/rdf/m/07d5b)

<http://schema.org/birthDate> "1955-6-8".
(http://www.imdb.com/name/nm3805083)
```

Figure 1: Mock-up of a trustable knowledge panel (based on a Google screenshot).

Let us assume, we want to state that the entity "Tim Berners-Lee" (TimBL) has "Web Developer" as an occupation. The following three sets of triples transmit the claim at different levels of granularity:

```
(1) [ex1:TimBL ex1:occ "Web Developer"]
(2) [ex2:TimBL ex2:job ex2:webDev]
(3) [ex3:TimBL ex3:work ex3:work42],
    [ex3:work42 ex3:occ ex3:webDev],
    [ex3:work42 ex3:since "1989-03"]
```

In (1), only a non-clickable string would be displayed for "Web Developer". With (2) and (3), a link to `ex2:webDev` or `ex3:webDev` can be provided where potentially more information about the profession can be retrieved. However, if we also want to model "since when Tim Berners-Lee has been a Web Developer", we make use of n-ary[3] relations as shown in (3). In the example, we create an individual connecting node (`ex3:work42`) to combine the information that "Tim Berners-Lee has been a Web Developer since March 1989". While some vocabularies (such as schema.org[4] or the Open Graph Protocol[5]) commonly use the more coarse-grained variants of (1) and (2) in their modeling, web knowledge bases such as Freebase [4] (that has been discontinued) and Wikidata [33] enable fine-grained modeling with n-ary relations (context/qualifiers) as exemplified in (3). In general, it is the authors' decision which level of detail they want to address with the data they publish on the web. Our entity data fusion approach performs the complex alignment of different vocabularies and automatically moves similar claims—expressed RDF triple(s)—into the same clusters.

[2] See also Tim Berners-Lee's idea of the "Oh yeah?" button – https://www.w3.org/DesignIssues/UI.html#OhYeah.

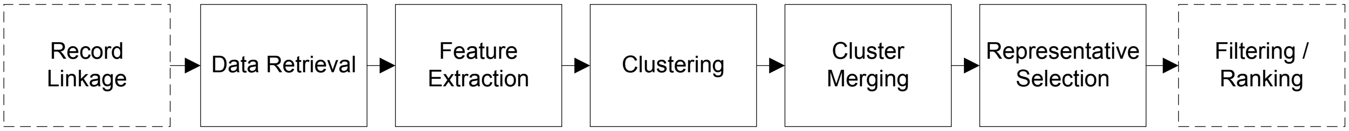


Figure 2: Overview: entity-centric data fusion.

3 APPROACH

Our approach for entity-centric data fusion starts with a URI $u \in U$ (with U being the set of all URIs) that identifies a specific focus entity e and returns a filtered and ranked set of claims about e. Different resources can provide data for an entity, possibly under different URIs. Our approach uses a pipeline which consists of seven different processing steps that are visualized in Figure 2:

(1) Record Linkage: discover the URIs of all resources equivalent to the entity e;
(2) Data Retrieval: retrieve RDF data for each resource and its connected resources;
(3) Feature Extraction: extract a set of information compounds (that we call “path features”) from RDF data;
(4) Clustering: run agglomerative hierarchical clustering on the set of path features;
(5) Cluster Merging: refine clusters by merging;
(6) Representative Selection: identify claims as cluster representatives;
(7) Filtering/Ranking: use different cluster features for filtering or ranking.

The main focus is on the steps 2 to 6 but we also provide general information on step 1 (i.e., what kind of input do we expect from the record linkage step) and step 7 (i.e., what kind of output does the pipeline produce and how can the results be used). We now describe each step of the pipeline in detail.

3.1 Record Linkage

The topic of record linkage has a long tradition in statistics and different subfields of computer science, including databases and information retrieval [22]. The main idea is to retrieve different files, entries, or identifiers that refer to the same entity (e.g., a specific person). The problem has also been explored in the (Semantic) web context [1, 18, 19]. With the use of explicit equivalence (e.g., by using `schema:sameAs` or `owl:sameAs`), the availability of a variety of algorithms (e.g., [15] for a recent work), and the availability of systems that offer record linkage as a service (e.g., sameAs.org), we regard this problem as sufficiently addressed. The record linkage approach is expected to take one URI for an entity e as input (e.g., http://dbpedia.org/resource/Tim_Berners-Lee) and then produces an extended set R of reference URIs that all refer to e, for example:

$$R = \{ \quad \texttt{ex1:TimBL, ex2:TimBL} \quad \}$$

3.2 Data Retrieval

We assume that all structured data is available as RDF [23]. The sources either directly provide RDF in N-Triples, Turtle, RDF/XML or JSON-LD via HTTP content negotiation or provide HTML pages with embedded markup, where RDFa, Microdata, and JSON-LD are the most supported formats [16, 24]. An RDF graph is defined as follows:

Definition 3.1 (RDF Graph, RDF Triples). With the three sets of URIs U, blank nodes B, and literals L, an RDF graph G is defined as:

$$G \subseteq (U \cup B) \times U \times (U \cup B \cup L) \tag{1}$$

The elements $t \in G$ of a graph G are called triples. The first element of a triple t is called the “subject”, the second “predicate”, and the third “object”. URIs provide globally unique identifiers; blank nodes can be used instead of URIs if there is no URI available for an entity, or the entity’s URI is unknown; and RDF literals encode data type values such as strings or integers.

For each reference URI $r \in R$ from the record linkage step, we aim to retrieve RDF data. If one of the URIs offers RDF, the crawler performs a breadth-first search around the URI (up to a certain depth d). For example, if the triple `[ex2:TimBL ex2:job ex2:webDev]` is contained in the retrieved dataset of `ex2:TimBL`, the crawler also tries to retrieve RDF data from the URI `ex2:webDev`. In addition, the crawler also retrieves information about the used predicates; in this case the crawler retrieves data from `ex2:job`. During this process, the crawler stores the complete path to the finally delivering URI (i.e., the URI that returns data with status 200 in case of redirects) for each request. That URI is used as a graph name for all retrieved corresponding triples (therefore, with the notion of named graphs – see Definition 3.2, producing quads). Cycles in the breadth-first search are resolved if the URI has already been requested in the same search around $r \in R$ or if the target URI is contained in the set R (or their respective redirect variants; i.e., cross-references in R are removed in this step). In these cases, we do not retrieve the URI a second time. The result of the data retrieval step is an RDF dataset that contains a forest of trees that each have one reference URI as a root (together with provenance information, that is, the URIs of the graphs in which the RDF triples occur). Figure 3 shows an example for such a forest.

Overall, the forest together with the provenance information forms a set of RDF graphs (i.e., an RDF dataset). An RDF dataset can cover multiple RDF graphs and is defined as follows:

Definition 3.2. [Named Graph, RDF Dataset] Let D be the set of RDF graphs and U be the set of URIs. A pair $\langle d, u \rangle \in D \times U$ is called a named graph. An RDF dataset consists of a (possibly empty) set of named graphs (with distinct names) and a default graph $d \in D$ without a name.

[3] “Defining N-ary Relations on the Semantic web” – http://www.w3.org/TR/swbp-n-aryRelations
[4] schema.org – http://schema.org
[5] Open Graph Protocol – http://ogp.me/

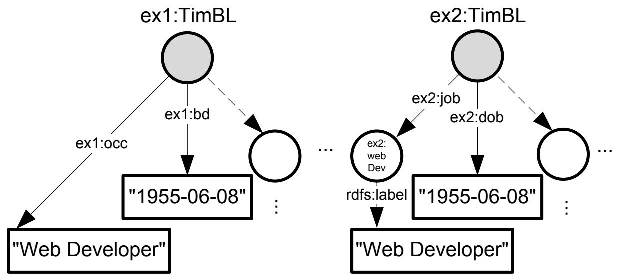


Figure 3: Output of data retrieval: an RDF dataset containing a forest of trees, each with a reference URI as a root.

3.3 Feature Extraction

We produce path features from the forests in the RDF dataset D created in the data retrieval step. In the following, we consider paths in the tree from the root to a leaf. In accordance to the definition of RDF, each tree can have two types of leaves: URI nodes or literal nodes. However, leaves that are URI nodes do not provide sufficient information as the node itself is a URI that was not retrieved. The system only knows that it exists. For example, if we crawl `ex2:TimBL` only with depth 0 (i.e., `ex2:TimBL` and predicate URIs are retrieved) the system knows that the node `ex2:webDev` exists but we do not get the label "Web Developer" if the URI is not retrieved (i.e., it is not in the RDF Dataset D as a graph name). Therefore, we only consider paths that end with a literal node.

Definition 3.3 (Path Feature). Let G be an RDF graph. A path feature p is a sequence of triples in G. A path feature fulfills the following three conditions:

(1) It starts with a triple $t_1 \in G$ of the retrieved graph G that has a reference URI in the subject position.
(2) It terminates with a triple $t_2 \in G$ of the retrieved graph G that has a literal in the object position.
(3) It does not contain triples that have a reference URI in the object position (to avoid loops).
(4) If it contains two consecutive triples $t_1, t_2 \in G$, the object of t_1 needs to be the subject of t_2.

Single triples that fulfill the above conditions are also path features. We refer to path features that involve multiple triples as "multi-hop" path features and to those that are constituted by only one triple as "single-hop" path features. We use the ⧟ symbol to denote the sequence of path features.

In our example, if we only consider the depicted claims of Figure 3, the following path features can be identified:

(1) `[ex1:TimBL ex1:bd "1955-06-08"]`
(2) `[ex1:TimBL ex1:occ "Web Developer"]`
(3) `[ex2:TimBL ex2:dob "1955-06-08"]`
(4) `[ex2:TimBL ex2:job ex2:webDev]`⧟
`[ex2:webDev rdfs:label "Web Developer"]`

Next, we introduce a way that enables us to represent path features as linked lists of strings: we remove all URI nodes and use the `rdfs:label` of the predicate URIs. A predicate can also have more than one label in the same language, so that we create a representation for each. For example, `ex2:job` may have the additional label "profession". To take this into account, we add another string representation for the full path feature. For all text-based literals and labels we fix the language. In practice, best results can be achieved with English as vocabularies often provide labels only in that language.

We collect all string representations in a multi-valued map M:

```
M = [ ("birth date"→"1955-06-08", ⟨1⟩) ;
("occupation"→"Web Developer", ⟨2⟩) ;
("date of birth"→"1955-06-08", ⟨3⟩) ;
("occupation"→"label"→"Web Developer", ⟨4⟩) ;
("profession"→"label"→"Web Developer", ⟨4⟩) ]
```

3.4 Clustering

We cluster path features in accordance to their string representations. At this point, a key feature of the approach—the *entity centricity*—mitigates the occurrence of ambiguities and unwanted merges. For example, the string "web" has only one reasonable meaning in the vicinity of the entity `ex1:TimBL` while in the whole web graph there are many different meanings for the term.

Similarity. In order to compare the string representations with each other, we use string similarity functions as they are proposed for ontology alignment [7]. For two given string representations we compare the head h[6] (i.e., the label of the first predicate) and the tail t (i.e., the label of the leaf node) of each string representation $l_i \in keySet(M)$ respectively. We compute the common result with a linear combination ($0 \le \lambda \le 1$):

$$\mathrm{sim}(l_1, l_2) = \lambda \cdot \mathrm{sim}(h(l_1), h(l_2)) + (1 - \lambda) \cdot \mathrm{sim}(t(l_1), t(l_2)) \quad (2)$$

In our experiments we set $\lambda = 0.5$, which produced good results for the clustering. The string similarity function incorporates tokenization (to) and normalization steps. We distinguish between single-token and multi-token strings:

$$\mathrm{sim}(s_1, s_2) = \begin{cases} \mathrm{jw}(s_1, s_2) & \text{if } |to(s_1)| = 1 \\ & \& \ |to(s_2)| = 1 \\ \mathrm{ja}(to(s_1), to(s_2)) & \text{otherwise} \end{cases} \quad (3)$$

Single-token strings use the Jaro-Winkler similarity metric (jw) and multi-token strings use Jaccard similarity (ja). These measures are recommended in [7] for achieving high precision. For both string similarity measures, a value of 0 means no similarity and 1 is an exact match.

Clustering. We compute a similarity matrix for all string representations as an input for agglomerative hierarchical clustering. The clustering is based on two steps: in the beginning, the linkage of all elements is computed and afterwards the clusters are formed by a cut-off. The linkage starts with clusters of size 1 and uses the similarity matrix in order to link two clusters. This is done in accordance to the smallest Euclidean distance of any two elements in the respective clusters. In the matrix, the elements are represented as column vectors. We repeat this step until all clusters are linked. The

[6] We assume that the first predicate is commonly more descriptive than the second or third predicate with respect to the focus entity e. It has to be noted that, in some cases, the second or third predicate could make a better fit for string comparison. For example, a string representation for the n-ary "work"-relation (see Example (3) in Section 2) could be: "work"→"occupation"→"label"→"Web Developer".

linkage is then used to determine a cut-off level that produces n or fewer clusters. Under the assumption that all resources in R provide RDF data and that each covers the same amount of information, the value of n can be set to $\left\lceil \frac{|M|}{|R|} \right\rceil$.[7] In our running example n would be $\left\lceil \frac{5}{2} \right\rceil = 3$. After the clustering, we use the map M to move back from the string representation level to the path feature level. The clusters are then represented as follows:

- **Cluster 1:** `{ [ex1:TimBL ex1:bd "1955-06-08"], [ex2:TimBL ex2:dob "1955-06-08"] }`
- **Cluster 2:** `{ [ex1:TimBL ex1:occ "Web Developer"], [ex2:TimBL ex2:job ex2:webDev]○–○ [ex2:webDev rdfs:label "Web Developer"] }`
- **Cluster 3:** `{ [ex2:TimBL ex2:job ex2:webDev]○–○ [ex2:webDev rdfs:label "Web Developer"] }`

In accordance to the defined similarity measure, the items of Cluster 2 have a perfect match. The items of Cluster 1 have a high similarity as the literal values match perfectly and the predicates have a partial match. The most dissimilar item is Path Feature 4 with its alternative label "profession" for `ex2:job`. This item ends up in its own cluster (as the number of total clusters is predefined with 3, see above).

3.5 Cluster Merging

After the clustering, similar string representations of path features are in the same cluster but some information is also dispersed. For example, Cluster 2 and Cluster 3 represent similar information. The data retrieval step (see Section 3.2) also retrieves path features that include information about related entities. For example, in the case of `ex1`, if we also cover the birth place of the entity "Tim Berners-Lee", via `[ex1:TimBL ex1:bp ex1:London]` we produce a lot of path features that differ only in factual information about London. `ex2` might cover similar claims and its information about London might be gathered in the same clusters as the claims from `ex1`. This naturally leads to many clusters that have the following shape:

```
{ [ex1:TimBL ex1:bp ex1:London]○–○
[ex1:London ex1:long "-0.127"],
[ex2:TimBL ex2:pob ex2:London]○–○
[ex2:London ex2:longitude "-0.1275"] }
```

Similar clusters would be formed about the latitude of London, its population, total area, etc. In the case of the entity "Tim Berners-Lee", another fraction of different clusters would cover claims about the MIT (e.g., number of students, founding year, etc.). A commonality among these fractions (e.g., London, MIT) of clusters is that the first triples of the contained path features are overlapping with the first triples of the path features in other clusters. The individual entity focus (in the example "Tim Berners-Lee") provides that only these first triples are relevant as—independent of the modeling granularity—the first hop is most relevant to the entity. Therefore, we can merge clusters in which the first triples of the path features are overlapping.

In our example, the first triples of Cluster 2 and Cluster 3 are as follows:

- **Cluster 2:** `{ [ex1:TimBL ex1:occ "Web Developer"], [ex2:TimBL ex2:job ex2:webDev] }`
- **Cluster 3:** `{ [ex2:TimBL ex2:job ex2:webDev] }`

For the merging we apply the following method: if, in terms of first triples, two clusters have an equal or higher degree of overlap (estimated via Jaccard index, that has a range between 0 and 1) than a threshold ϵ,[8] the clusters are merged. Note that the criteria for merging clusters is based on structure (i.e., first triples of path features) and the measure with which we derive the clusters is string similarity.

In our example, with $\epsilon = 0.5$, Cluster 2 and Cluster 3 are merged:

Cluster 2: `{ [ex1:TimBL ex1:occ "Web Developer"],`
```
[ex2:TimBL ex2:job ex2:webDev]○–○
[ex2:webDev rdfs:label "Web Developer"],
[ex2:TimBL ex2:job ex2:webDev]○–○
[ex2:webDev rdfs:label "Web Developer"] }
```

Clusters containing `[ex1:TimBL ex1:bp ex1:London]` as a first triple would also get merged. While first triples of single-hop path features such as `[ex1:TimBL ex1:occ "Web Developer"]` can occur only in multiple clusters if there are more labels for the predicate, multi-hop path features can generate a variety of different label-leaf combinations for their string representations and the first triple or—like in the example—the complete path feature can occur in multiple different clusters before the merging step. In our approach, the combination of path features, their clustering, and the merging of clusters can address all of these cases in a suitable manner.

3.6 Representative Selection

For each cluster, we can select two types of representatives: one general representative and one representative for each source. Both types of representatives are needed for the scenario of Figure 1: one triple to be shown in the panel and one triple per source to support the presented triple. Before we present the details of the representative selection approach, we need to define the term "source". For this we tracked the provenance of each triple in the data retrieval step (cf. Section 3.2). For a specific path feature, we take the first triple: the hostname of the delivering URI of this triple is considered as the *source* of the path feature. The complete delivering URI of a source representative may be used for a more detailed output (as exemplified in Figure 1).

Cluster representative. We consider two cases for the cluster representative:

(1) If the cluster contains only single-hop path features, return the triple that has the highest similarity (see Formula 2) to all other triples.
(2) If the cluster contains only multi-hop path features or single-hop and multi-hop path features use the first triple of each multi-hop path feature and count its occurrence

[7] R is defined as the set of all reference URIs in Section 3.1, M is defined as the multi-valued map between string representation and the respective path features in Section 3.3.

[8] The value of ϵ is flexible and can be adjusted within the range of 0 and 1.

in the cluster. The first triple that occurs most often in the cluster is returned as the representative.

In our example, the first case returns any of the two birth-date triples (as they have equal similarity to each other) for Cluster 1. The first case enables to select the most common representation among multiple candidates. For example, Wikidata provides also "label"→"Sir Tim Berners-Lee" for the entity and the according path feature gets clustered together with the path feature represented by "label"→"Tim Berners-Lee" from Wikidata[9], IMDb, Freebase etc. The first case selects the representative that is most similar to all others and chooses the version without "Sir".

In our example, the second case returns [ex2:TimBL ex2:job ex2:webDev] as a representative for Cluster 2 (the triple occurs twice). The idea of the second case is that links to other resources (multi-hop) are always better than returning a plain string (single-hop). However, the single-hop path features in multi-hop clusters support the respective claim as a source. In addition, the second case returns a triple that occurs in most path features and, as such, the linked resource (i.e., ex2:webDev in the example) can provide most information on the claim that is described by the cluster.

For the running example, the output of the representatives would be as follows:

```
[ex1:TimBL ex1:bd "1955-06-08"],
[ex2:TimBL ex2:job ex2:webDev]
```

For both claims, the two sources ex1 and ex2 can be provided as references.

Source representative. Source representatives are selected in the same way as the cluster representative with the following restriction: it is chosen as (1) the most similar or (2) most often occurring representative from a single source (e.g., dbpedia.org) compared to all entries across sources.

3.7 Filtering / Ranking

Our approach covers the clustering of similar claims about entities. It does not address steps that can build on the gained information. In this section we provide an overview.

An important aspect, that we have not yet addressed, is the handling of contradicting information. In general, following the open-world assumption, we consider all made claims of all sources as true. If a claim is missing in one source but occurs in another, it can be true. If, in the case of persons, different sources provide different claims about spouses, employers, and even the birth dates, we consider all of them as true. However, as a general idea, we assume that claims are more likely to be true if they are made by multiple different sources.[10] In fact, the more sources support a claim, the more likely it is to be valid or important. In contrast, if a claim is made only by a single source, it is considered less likely or unimportant. The lack of (a sufficient amount of) sources and the explanations why certain claims are provided in knowledge panels has led to criticism [6, 14]. With the presented entity data fusion approach, we can support the identification of additional sources for claims. This enables users to verify the individual sources and decide themselves whether they want to trust the claim or not. In addition, in order to enable an automatically produced trustability score, additional measures—such as PageRank [5] or knowledge-based trust [12]—can be applied on the sources for each claim.

In a similar way, additional support for the notability of claims can be estimated: the more sources support a claim about an entity, the more it can be considered as important. This is in line with the ideas of [32] that present entities in this manner (ranking claims by the number of sources that support them).

4 EXPERIMENTS

In our experiments, we evaluated our entity data fusion method relative to the Sig.ma baseline established in [32]. We compare the coverage and the number of sources with respect to the scenario of a trustable knowledge panel (see Section 1). The idea is that we do not want to compare agreement on randomly selected claims but to make sure that the evaluated claims would actually be presented to an end user. For this, we use the claims presented in the Google Knowledge Graph (GKG) panels. With regard to the size and the heterogeneity of the dataset (actual data from the web), this restriction made the task of evaluation feasible.

4.1 Dataset

The TREC entity track was last run in 2011.[11] We used the provided evaluation data from that year[12] and selected the entity names of the REF and ELC tasks. This produced 100 entities with two duplicates. Afterwards, we tried to identify the DBpedia URIs for the remaining set of 98 entities. For 18 entities (e.g, "Landfall Foundation" or "Foundation Morgan horses") we could not find according DBpedia identifiers (and also Google did not provide a graph panel for these entities). Therefore, the final set of entities contained 80 entities. This included persons, organizations, universities, places, bands, etc. The service sameAs.org then enabled the retrieval of the according Freebase identifiers (e.g., m/027bp7c – entity "Abraham Verghese") and we could then retrieve Google summaries by adding the GKG API namespace http://g.co/kg/ to these IDs, for example http://g.co/kg/m/027bp7c. We manually retrieved GKG panels by storing the respective HTML to files. In this context, we used http://google.com in English language with a clean browser history for each entity.

The list of the used entities, their DBpedia and Google identifiers, the crawled dataset, the stored Google result pages, and the output of our approach are available at http://people.aifb.kit.edu/ath/entity_data_fusion.

4.2 Baseline: Sig.ma

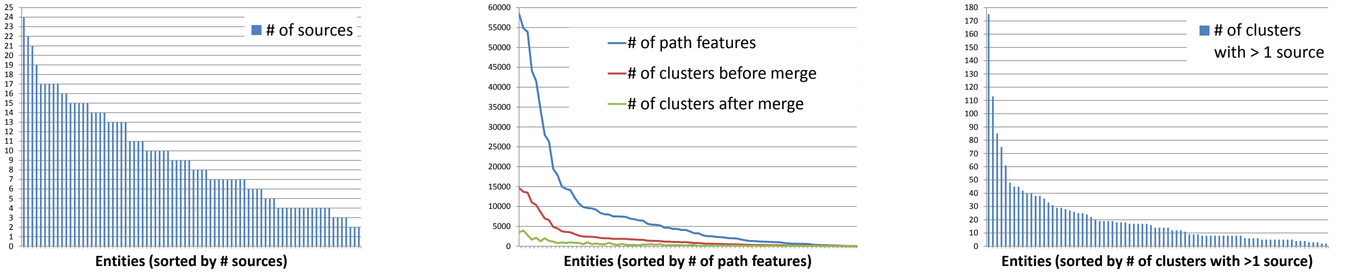
The Sig.ma system described in [32] provides basic functionality on entity data fusion. The approach is mostly based on string modification in order to derive a uniform representation. In particular, the provided URIs for properties and the URIs/literals for values are analyzed heuristically. The approach cannot deal with n-ary relations and can only rudimentary reconcile between 0-hop and 1-hop granularity levels. However, in these cases it can serve as

[9] Wikidata provides multiple English labels for this entity.

[10] Note: In a web setting, this assumption is not necessarily correct as the sources are often not independent from each other. We discuss this matter in Section 4.6.

[11] TREC tracks – http://trec.nist.gov/tracks.html

[12] TREC entity track 2011 – http://trec.nist.gov/data/entity2011.html



(a) Number of different sources for each entity.

(b) Number of path features and clusters before/after the merging step.

(c) Number of clusters with more than one source.

Figure 4: Different statistics on the distribution of path features, clusters, and sources. The ticks on the x-axes each represent one entity of the TREC dataset.

a baseline so we re-implemented the main ideas[13] of Sig.ma by performing the following steps:

(1) We use the properties and values of triples where an identifier for an entity is involved, for example:
`ex4:occupation "Web Developer"@en`

(2) For URIs (in the property or object position) we use the last segment of the URI (e.g., `occupation`). Typical patterns such as *camelCase* and dashes/underscores are split up. Literal values are used without further modification. All strings are transformed to lower case. For the reconciliation, the Sig.ma approach does not make use of `rdfs:label` [32].

(3) These basic string representations are then aggregated with an exact match and by attributing their sources:
`"occupation Web Developer"`
`(http://example4.com, http://example5.com)`

4.3 Configuration

We applied the two data fusion methods on 80 entities of the TREC entity dataset. We used the `sameAs.org` service as a record linkage approach with the DBpedia identifiers as an input. Multiple crawls were performed in order to account for periods of unavailability of resources. The crawls happened in June 2015. The crawler operated with depth 1 and retrieved RDF data via content negotiation. After the individual crawls were completed the retrieved data was merged. Per entity, there were 2 to 24 different sources while 75% of the entities included RDF information from at least 5 sources (see Figure 4a). For our method, for each entity, we computed the similarity matrix of the English string representations of all path features. We set the parameter $\lambda = 0.5$. For this matrix, we produced the linkage and retrieved $n = \left\lceil \frac{|M|}{4} \right\rceil$ clusters for each entity. We merged all clusters at an overlap threshold of $\epsilon = 0.5$. An overview of the distribution of the numbers of path features, clusters, and merged clusters is provided in Figure 4b. All entities had more than two clusters with at least two sources and 59% of the entities had more than 10 such clusters. An overview of this distribution is provided in Figure 4c.

4.4 Evaluation Setup

The evaluation included two steps, the matching of GKG claims to clusters of the output of the respective systems and the evaluation of the identified matches.

Step 1: Match GKG claims to clusters. For the evaluation of the quality of the results, the Google result pages and the produced output of the systems needed to be aligned. Unfortunately, although the data presented by Google is often found in Freebase (which was covered by our crawl), it was not possible to identify a sufficient number of direct links. On the one hand, this was due to the incorrect Turtle RDF output produced by Freebase. On the other hand, a lot of information covered by Freebase included n-ary relations that are presented flat in GKG panels. Therefore, starting from a GKG claim, it is difficult to determine the respective Freebase claim—especially if a variety of domains are covered (as it was the case for the TREC entities). As a consequence, we nominated two human evaluators (both experts on RDF and related technologies) and asked them to provide a manual matching. For all entities, the following was performed: For each claim that was presented in the GKG panel, they used the systems' output to identify clusters in which at least one source representative matched the information content of the GKG claim.

Step 2: Evaluation of matches. For all clusters in the output of the systems that matched a specific GKG claim, the evaluators were instructed to choose the cluster that had most correct sources (i.e., clusters where most source representatives match the information content of the GKG claim). The number of correct sources of this cluster was then documented. In the same step the evaluators kept track of the following two types of error:

Type 1 error: Number of source representatives in the best-fit cluster that did not match the GKG claim (false positives).

Type 2 error: Number of source representatives in other clusters, that also matched the information content of the GKG claim (false negatives).

4.5 Evaluation Results

The evaluators identified 755 claims in the GKG panels of the 80 TREC entities. In average, each GKG panel covered 9.4 claims. Table 1 respectively present the main results of our approach and

[13] We omitted several highly customized rules of Sig.ma such as the "[...] manually-compiled list of approximately 50 preferred terms" [32].

Table 1: Results for our approach and Sig.ma: the number of produced GKG claims, GKG coverage, number of type 1 errors, number of type 2 errors, precision, recall, and f-measure at different thresholds for the number of sources. The # symbol should be read as "number of".

# sources in output:	≥ 1	≥ 2	≥ 3	≥ 4	≥ 5
Our approach:					
# GKG claims:	414	235	135	76	39
GKG coverage:	55%	31%	18%	10%	5%
# type 1 errors:	81	46	26	17	12
# type 2 errors:	146	81	43	26	16
Precision:	0.84	0.84	0.84	0.82	0.76
Recall:	0.74	0.74	0.76	0.75	0.71
F-measure:	0.78	0.79	0.80	0.78	0.74
Sig.ma:					
# GKG claims:	299	112	70	44	9
GKG coverage:	40%	15%	9%	6%	1%
# type 1 errors:	0	0	0	0	0
# type 2 errors:	304	151	92	57	34
Precision:	1.0	1.0	1.0	1.0	1.0
Recall:	0.50	0.43	0.43	0.44	0.21
F-measure:	0.66	0.60	0.60	0.61	0.35

Table 2: Statistics about the 15 most occurring predicates (with respect to the 755 claims of the GKG panels) and according statistics for the number of (true-positive) sources each system provides (↓ min, ø avg, ↑ max).

Predicate	Count	Our approach			Sig.ma		
		↓	ø	↑	↓	ø	↑
label	80	0	3.18	7	1	3.29	7
abstract	80	0	1.74	3	0	0.99	2
founder	42	0	0.79	4	0	0.02	1
place of interest	26	0	0.00	0	0	0.00	0
location	25	0	1.28	4	0	0.24	1
subsidiary	24	0	0.13	1	0	0.04	1
phone number	24	0	0.00	0	0	0.00	0
book	23	0	0.39	1	0	0.09	1
college/uni.	22	0	0.00	0	0	0.00	0
longitude	21	0	3.24	8	1	1.90	4
latitude	21	0	3.10	8	1	1.95	4
ceo	20	0	0.35	5	0	0.10	1
alumni	18	0	0.00	0	0	0.00	0
founding date	15	0	0.93	2	0	0.80	1
founding year	15	0	2.00	6	0	0.93	2

Sig.ma. Our entity data fusion method produced 414 GKG claims (with a respective coverage of 55%) and, in total, 923 source representatives. The baseline Sig.ma produced 299 GKG claims (with a respective coverage of 40%). In almost all cases our approach outperforms Sig.ma by ×2 or higher with respect to the task of retrieving multiple sources per GKG claim (GKG coverage at ≥ 2, ≥ 3, etc.). Sig.ma only considers direct 1:1 matches which means that it produces a precision of 1.0 (there are no type 1 errors). As a side effect, this also implies a strongly reduced recall (which stems from the high number of type 2 errors). The recall levels of Sig.ma drop strongly when more than five sources are needed. In contrast, our approach produces high precision and recall levels and also remains stable when more sources are required (the small increases/decreases are due to the varying proportion of type 1/2 errors with respect to the respective coverage). These scores are also reflected in the respective f-measure scores where our approach outperforms Sig.ma by differences from 0.12 (≥ 1 source) up to 0.39 (≥ 5 sources). In only 22 cases out of 755, Sig.ma produced more sources than our approach. In these cases, relevant claims ended up in larger clusters that had different representatives chosen. Table 2 presents the 15 most-used predicates of the 755 GKG claims and the minimum, average, and maximum number of sources per claim for each of the two systems. It shows that there exist GKG claims (such as phone number or places of interest) that were not covered by any of the web data sources. This explains the gap between 755, the total number of GKG claims, and 414, the number of claims for which we could identify at least one source. In average, in almost all cases, our entity data fusion approach provides more sources than Sig.ma for all different claim predicates.

4.6 Discussion

The results of the experiments demonstrate the effectiveness of our entity data fusion approach. They show, that the recall is significantly improved by considering multiple granularity levels and by the approximate matching via string similarity. As a matter of fact, these factors affect the precision in a negative way, however (as the f-measure scores demonstrate) only to a point where the advantages of the improved recall have a significant overweight. In applications where precision is of ultimate importance, we would suggest an approach that utilizes direct or manually defined mappings. In the presented scenario of a trustable knowledge panel, we suggest to use our entity data fusion approach (which provides a highly improved recall).

A number of challenges that we encountered deal with the quality of Linked Data data on the web in general: not every URI is dereferenceable, not every URI provides RDF data, not all returned RDF data is in (any) correct format, not all RDF data contains information about the retrieved URI, not all RDF data contains labels, and not all RDF data contains language tags. We still made use of all these features and were able to retrieve RDF data from a number of reference URIs (up to 24) via content-negotiation and could make sufficient use of the provided data. For production environments, we would recommend the implementation of a data curation infrastructure that deals with the mentioned challenges.

RDF triples are often used in the subject-predicate-object style but, although—technically—the predicate provides a direction, every such triple also provides information about the object. Tim Berners-Lee encourages RDF creators not to put too much emphasis on the direction of RDF triples.[14] However, only few sources (DBpedia is one of them) provide information about an entity when it is in the object position of a triple. One way to address this matter could be to perform a full web crawl and apply path feature extraction also for triples that use the entity URI in the object position.

[14] Tim Berners-Lee: "Backward and Forward links in RDF just as important" – http://dig.csail.mit.edu/breadcrumbs/node/72

For a variety of parameters of the method, potential extension and optimization with a gold standard is possible. One particular point is literal/object similarity: Many literals are annotated by their type. For example, a birth date like "1955-06-08" often has xsd:date as data type. Therefore, additional (or alternative) similarity measures could be defined for the most common data types. Ultimately, this could be extended towards media similarity for URIs that represent an audio file, an image, or a video.

With increased crawling depth, the number of path features grows exponentially. As we compare path features via their string representation, and we have $|M| \cdot (|M| - 1)/2$ comparisons, this leads to a significant demand for computation time. One solution that we consider in order to mitigate this effect is locality-sensitive hashing [21]. This hashing method moves similar strings to similar buckets and strongly reduces the number of candidates for traditional string comparison.

One aspect that is not addressed in this work is the question "how can we verify that the sources gathered their information *independently* from each other?" Unfortunately, for small information units, such as triples, it is often impossible to gain a deep understanding of provenance if respective information is not explicitly given; especially if the claims are commonly known and true. A related task was addressed in [9] where the authors tackle the problem of copy detection by tracking different datasets and their change over time.

5 RELATED WORK

Our approach is most related to the data fusion and presentation method of Sig.ma by Tummarello et al. [32]. Sig.ma presents a rule-based, entity-centric data fusion method embedded in the context of semantic search. As such, further components of Sig.ma include object retrieval via keyword queries, parallel data gathering, live consolidation, and presentation. The presented entity data fusion approach is strongly focused on efficiency and relies on meaningful URIs, a frequently used feature of many vocabularies and datasets. In contrast, in our approach we fully rely on rdfs:label and can also deal with multiple languages and opaque identifiers as they are used in Wikidata or schema.org (that makes strong use of blank nodes). Although n-ary relations are mentioned in [32], they are not addressed by Sig.ma. In contrast, we designed our approach to deal with claims distributed over multiple hops and enable to align sources with different modeling granularities.

Data/Knowledge fusion: Recent work of Dietze points out the main challenges of "retrieval, crawling and fusion of entity-centric data" [8]. The author mentions the issues of missing (*owl:sameAs*) links, redundancy, and quality. In our work, we extend on that and lay particular focus on modeling granularities and introduce a feasible solution for the presented challenges. In [11], Dong et al. define knowledge fusion as the problem of constructing a large knowledge base from unstructured data (like HTML tables or natural language text) with different extractors from different sources. In contrast, data fusion is defined as the processing of a source-feature matrix for each entity where the entries mark the actual values. Our work lies between these two extremes as we deal with data for which we do not need extractors but the complexity of the data goes beyond database-like tables as we need to deal with different identifiers, vocabularies, and different modeling approaches. The work on knowledge-based trust by Dong et al. [12] is also related to our task. The authors estimate the trust-worthiness of web sources by extracting information and verifying its correctness. With this method, a trust value is computed for each web source. In contrast, we try to identify multiple occurrences of the same or similar claim. The methods complement each other and we could use the approach of Dong et al. [12] to compute the trustworthiness of the sources that we provide in our output.

Schema/Ontology alignment: The field of schema and ontology alignment has been very active in the past decade. Most relevant to our work is the approach by Suchanek et al. [29], that integrates relations, instances, and schemas. The authors use a probabilistic model to integrate each of the mentioned aspects. The approach is tested with YAGO, DBpedia, and IMDb. In contrast, in our work, we account for different granularities at the modeling level and also match claims that include more than one hop. Further, we test our approach in a real-world scenario with data from the web. The authors of [20] investigate on the problem of the large amount of different vocabularies. They state the question: "How Matchable Are Four Thousand Ontologies on the Semantic Web?" Although we do not explicitly deal with the merging of different vocabularies, our clustering approach could be used to mine complex mapping rules for vocabulary terms via patterns from different clusters (across entities) in an iterative way.

6 CONCLUSIONS

We have introduced a novel entity-centric approach for fusing claims from multiple web sources. Our approach works without any prior knowledge about the used vocabularies and just uses core features of the RDF data model. We have demonstrated two key features of the approach: the entity centricity, which enables the application of string similarity measures for clustering, and the robustness of the approach against fine- or coarse-grained RDF data modeling (via path features). In our experiments, we compared our system to the Sig.ma baseline and demonstrated that our system produces higher coverage, recall, and f-measure scores.

We also shed light on a variety of challenges that encompass the task of web-scale entity data fusion. In particular, the large number of different vocabularies, their individual modelling focus, and various issues with Linked Data quality bring additional complexity to an already computationally challenging problem.

We plan to address the use of existing mappings on the schema level based on rdfs:subClassOf, rdfs:subPropertyOf, owl:equivalentClass and owl:equivalentProperty. A strength of our current approach is that we do not need these mappings, as not many of them exist; schema.org, for example, only maps to a handful of external classes and properties. But we believe that, over time, more mappings will become available, either manually constructed or with the support of ontology alignment approaches that can handle schema diversity in arbitrary web data. In that line, we plan to extend the string-based similarity measure by a rule learning system that detects frequent vocabulary alignment patterns in the clusters and iteratively feeds this information back to the similarity measure. In the further work, we also plan to combine the presented approach with our entity summarization system LinkSUM [31].

ACKNOWLEDGMENTS

The research leading to these results has received funding from the Marie Curie International Research Staff Exchange Scheme (IRSES) of the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 612551 and by the German Federal Ministry of Education and Research (BMBF) within the Software Campus project "SumOn" (grant no. 01IS12051).

REFERENCES

[1] Krisztian Balog, David Carmel, Arjen P. de Vries, Daniel M. Herzig, Peter Mika, Haggai Roitman, Ralf Schenkel, Pavel Serdyukov, and Thanh Tran Duc. 2012. The First Joint International Workshop on Entity-oriented and Semantic Search (JIWES). *SIGIR Forum* 46, 2 (2012), 87–94. DOI:http://dx.doi.org/10.1145/2422256.2422268

[2] Tim Berners-Lee. 2006. Linked Data. https://www.w3.org/DesignIssues/LinkedData.html. (2006).

[3] Abraham Bernstein, James Hendler, and Natalya Noy. 2016. A New Look at the Semantic Web. *Commun. ACM* 59, 9 (2016), 35–37. DOI:http://dx.doi.org/10.1145/2890489

[4] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data.* ACM, New York, NY, USA, 1247–1250. DOI:http://dx.doi.org/10.1145/1376616.1376746

[5] Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30, 1 (1998), 107–117. DOI:http://dx.doi.org/10.1016/S0169-7552(98)00110-X

[6] Amy Cavenaile. 2016. You probably haven't even noticed Google's sketchy quest to control the world's knowledge. https://www.washingtonpost.com/news/the-intersect/wp/2016/05/11/you. (2016).

[7] Michelle Cheatham and Pascal Hitzler. 2013. String Similarity Metrics for Ontology Alignment. In *The Semantic Web – ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II.* Springer Berlin Heidelberg, Berlin, Heidelberg, 294–309. DOI:http://dx.doi.org/10.1007/978-3-642-41338-4_19

[8] Stefan Dietze. 2017. Retrieval, Crawling and Fusion of Entity-centric Data on the Web. In *Semantic Keyword-Based Search on Structured Data Sources: COST Action IC1302 Second International KEYSTONE Conference, IKC 2016, Cluj-Napoca, Romania, September 8–9, 2016, Revised Selected Papers*, Andrea Calì, Dorian Gorgan, and Martín Ugarte (Eds.). Springer International Publishing, Cham, 3–16. DOI:http://dx.doi.org/10.1007/978-3-319-53640-8_1

[9] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. 2009. Truth Discovery and Copying Detection in a Dynamic World. *Proc. VLDB Endow.* 2, 1 (2009), 562–573. DOI:http://dx.doi.org/10.14778/1687627.1687691

[10] Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge Vault: A Web-scale Approach to Probabilistic Knowledge Fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14).* ACM, New York, NY, USA, 601–610. DOI:http://dx.doi.org/10.1145/2623330.2623623

[11] Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Kevin Murphy, Shaohua Sun, and Wei Zhang. 2014. From Data Fusion to Knowledge Fusion. *Proc. VLDB Endow.* 7, 10 (2014), 881–892. DOI:http://dx.doi.org/10.14778/2732951.2732962

[12] Xin Luna Dong, Evgeniy Gabrilovich, Kevin Murphy, Van Dang, Wilko Horn, Camillo Lugaresi, Shaohua Sun, and Wei Zhang. 2015. Knowledge-based Trust: Estimating the Trustworthiness of Web Sources. *Proc. VLDB Endow.* 8, 9 (2015), 938–949. DOI:http://dx.doi.org/10.14778/2777598.2777603

[13] Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. 2014. Introducing Wikidata to the Linked Data Web. In *The Semantic Web – ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I.* Number 8796 in Lecture Notes in Computer Science. Springer International Publishing, 50–65. DOI:http://dx.doi.org/10.1007/978-3-319-11964-9_4

[14] Heather Ford and Mark Graham. 2016. *Code and the City.* Routledge, Chapter Semantic Cities: Coded Geopolitics and the Rise of the Semantic Web, 200–214.

[15] Anja Gruenheid, Xin Luna Dong, and Divesh Srivastava. 2014. Incremental Record Linkage. *Proc. VLDB Endow.* 7, 9 (2014), 697–708. DOI:http://dx.doi.org/10.14778/2732939.2732943

[16] Ramanathan V. Guha, Dan Brickley, and Steve MacBeth. 2015. Schema.Org: Evolution of Structured Data on the Web. *Queue* 13, 9, Article 10 (2015), 28 pages. DOI:http://dx.doi.org/10.1145/2857274.2857276

[17] Daniel Hernández, Aidan Hogan, and Markus Krötzsch. 2015. Reifying RDF: What Works Well With Wikidata?. In *Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems (CEUR Workshop Proceedings)*, Vol. 1457. CEUR-WS.org, 32–47.

[18] Daniel M. Herzig, Peter Mika, Roi Blanco, and Thanh Tran. 2013. Federated Entity Search Using On-the-Fly Consolidation. In *The Semantic Web – ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I.* Springer Berlin Heidelberg, Berlin, Heidelberg, 167–183. DOI:http://dx.doi.org/10.1007/978-3-642-41335-3_11

[19] Aidan Hogan, Andreas Harth, and Stefan Decker. 2007. Performing Object Consolidation on the Semantic Web Data Graph. In *Proceedings of 1st I3: Identity, Identifiers, Identification Workshop co-located with the 16th International World Wide Web Conference (WWW2007), Banff, Alberta, Canada.*

[20] Wei Hu, Jianfeng Chen, Hang Zhang, and Yuzhong Qu. 2011. How Matchable Are Four Thousand Ontologies on the Semantic Web. In *The Semantic Web: Research and Applications: 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Greece, May 29-June 2, 2011, Proceedings, Part I.* Springer Berlin Heidelberg, Berlin, Heidelberg, 290–304. DOI:http://dx.doi.org/10.1007/978-3-642-21034-1_20

[21] Piotr Indyk and Rajeev Motwani. 1998. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing (STOC '98).* ACM, New York, NY, USA, 604–613. DOI:http://dx.doi.org/10.1145/276698.276876

[22] Nick Koudas, Sunita Sarawagi, and Divesh Srivastava. 2006. Record Linkage: Similarity Measures and Algorithms. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data.* ACM, New York, NY, USA, 802–803. DOI:http://dx.doi.org/10.1145/1142473.1142599

[23] Frank Manola and Eric Miller. 2004. RDF Primer. (2004). W3C Recommendation, http://www.w3.org/TR/rdf-syntax/.

[24] Robert Meusel, Petar Petrovski, and Christian Bizer. 2014. The WebDataCommons Microdata, RDFa and Microformat Dataset Series. In *The Semantic Web – ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I.* Springer International Publishing, Cham, 277–292. DOI:http://dx.doi.org/10.1007/978-3-319-11964-9_18

[25] Rahul Parundekar, Craig A. Knoblock, and José Luis Ambite. 2010. *Linking and Building Ontologies of Linked Data.* Springer Berlin Heidelberg, Berlin, Heidelberg, 598–614. DOI:http://dx.doi.org/10.1007/978-3-642-17746-0_38

[26] Dominique Ritze, Christian Meilicke, Ondřej Šváb Zamazal, and Heiner Stuckenschmidt. 2009. A Pattern-based Ontology Matching Approach for Detecting Complex Correspondences. In *Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) Collocated with the 8th International Semantic Web Conference (ISWC 2009) (CEUR Workshop Proceedings)*, Vol. 551. CEUR-WS.org, 25–36.

[27] Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Natasha Noy, and Arnon Rosenthal (Eds.). 2009. *Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) Collocated with the 8th International Semantic Web Conference (ISWC 2009).* CEUR Workshop Proceedings, Vol. 551. CEUR-WS.org.

[28] Amit Singhal. 2012. Introducing the Knowledge Graph: things, not strings. http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html. (2012).

[29] Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. 2011. PARIS: Probabilistic Alignment of Relations, Instances, and Schema. *Proc. VLDB Endow.* 5, 3 (2011), 157–168. DOI:http://dx.doi.org/10.14778/2078331.2078332

[30] Andreas Thalhammer. 2016. *Linked Data Entity Summarization.* Phdthesis. KIT, Fakultät für Wirtschaftswissenschaften, Karlsruhe. DOI:http://dx.doi.org/10.5445/IR/1000065395

[31] Andreas Thalhammer, Nelia Lasierra, and Achim Rettinger. 2016. LinkSUM: Using Link Analysis to Summarize Entity Data. In *Web Engineering: 16th International Conference, ICWE 2016, Lugano, Switzerland, June 6-9, 2016. Proceedings.* Lecture Notes in Computer Science, Vol. 9671. Springer International Publishing, Cham, 244–261. DOI:http://dx.doi.org/10.1007/978-3-319-38791-8_14

[32] Giovanni Tummarello, Richard Cyganiak, Michele Catasta, Szymon Danielczyk, Renaud Delbru, and Stefan Decker. 2010. Sig.ma: Live views on the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web* 8, 4 (2010), 355–364. DOI:http://dx.doi.org/10.1016/j.websem.2010.08.003

[33] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85. DOI:http://dx.doi.org/10.1145/2629489

[34] Denny Vrandečić, Varun Ratnakar, Markus Krötzsch, and Yolanda Gil. 2011. Shortipedia: Aggregating and Curating Semantic Web Data. *Web Semantics: Science, Services and Agents on the World Wide Web* 9, 3 (2011), 334–338. DOI:http://dx.doi.org/10.1016/j.websem.2011.06.006